UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/710,356	07/02/2004	Steven T. Shaughnessy	BOR-012	4355

64107 7590 11/09/2009

KOKKA & BACKUS, PC
200 PAGE MILL ROAD
SUITE 103
PALO ALTO, CA 94306

EXAMINER	
SANDERS, AARON J	
ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
11/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/710,356	SHAUGHNESSY, STEVEN T.	
	Examiner	Art Unit	
	AARON SANDERS	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 August 2009*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *31-55* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *31-55* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Amendment

The amendment filed 20 August 2009 has been entered. Claims 31-55 are pending. Claims 31, 42, and 44 are currently amended. Claims 1-30 are cancelled. No claims are new. This action is FINAL.

Claim Objections

Claims 31 and 43 should have been marked "Currently Amended," not "Previously Presented." See 37 C.F.R. 1.121(c). The amendments have been entered as they appear to be a bona fide attempt.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 31-33 and 36-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Klein et al., U.S. 6,631,374 ("Klein"), in view of Ganesh et al., U.S. 6,192,377 ("Ganesh"), in view of Weems, "Shadow Cache," University of Massachusetts, May 2004 ("Weems"), and in view of Natarajan et al., "Log Sequence Numbers," University of Wisconsin, May 2003 ("Natarajan").

31. Klein teaches "*A method, comprising... collecting an active transaction list of all active transactions when a read-only transaction starts*," see Fig. 10 and col. 8, ll. 44-58, "For each ITE 51, transactions are rolled back in an ordered manner... Thus, if the transaction referenced by the transaction identifier xid is active (block 163), the entire transaction is rolled back to logically undo the transaction (block 164). If there are more active transactions (block 165), the rollback is repeated (block 165) until all active transactions have been rolled back."

Klein teaches "*loading a modified block into a read-only cache view*," see Fig. 10, step 161. Klein does not teach "*the modified block including a log sequence number*." Natarajan does, however, see 4.1, "Every log record is assigned a unique Log Sequence Number (LSN)... Also, the recovery subsystem will supply an LSN and expect the log manager to read the corresponding log record." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Natarajan's teachings would have allowed Klein's method to gain an easy way to compare records to determine if the recovery subsystem should redo an update, see Natarajan, 4.1.

Klein teaches "*performing physical redo operations or physical undo operations based upon the comparing as the modified block is loaded into the read-only cache view*," see Fig. 10, steps 166-67, where the claimed "comparing," when combined with Ganesh and Natarajan, would be the referenced step 166. Klein does not teach "*wherein the physical redo operations or the physical undo operations are not performed on the modified block when the log sequence number is less than the fist log sequence number*." Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches using the current version of the data block if the snapshot (i.e. the "log sequence number" when combined with Natarajan) is less

than the buffer refresh time (i.e. the "first log sequence number" when combined with Natarajan), see Fig. 2, steps 206 and 216. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to use Natarajan's LSNs as Ganesh's snapshot and buffer times and to use Ganesh's time comparisons because doing so would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1. Klein does not teach "*and the physical redo operations are performed on the modified block when the log sequence number is greater than the first log sequence number and less than the second log sequence number.*" Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches using another version of the data block if the snapshot (i.e. "log sequence number") is greater than the buffer refresh time (i.e. "first log sequence number"), see Fig. 2, steps 206 and 208. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention that Ganesh's step 208 could be redoing the transaction instead of using a different version, since it is a predictable variation, see KSR International Co. v. Teleflex Inc., 550 U.S. 398, 417, 82 USPQ2d 1385, 1396 (2007) ("If a person of ordinary skill can implement a predictable variation, § 103 likely bars its patentability"). Klein does not teach "*and the physical undo operations are performed on the modified block when the log sequence number is greater than the second log sequence number.*" Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches undoing changes when the snapshot (i.e. "log sequence number") is greater than the data buffer commit time (i.e. "second log sequence number") and the transaction is active, see Fig. 2, steps 210, 214, and 212. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to use Natarajan's LSNs as Ganesh's snapshot and buffer

times and to use Ganesh's time comparisons because doing so would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1.

Klein teaches "*processing the active transaction list to physically undo an incomplete action and to logically undo an active transaction to generate transactional consistency in the read-only cache view*," see Fig. 10 and col. 8, ll. 44-58, "Thus, if the transaction referenced by the transaction identifier xid is active (block 163), the entire transaction is rolled back to logically undo the transaction (block 164). If there are more active transactions (block 165), the rollback is repeated (block 165) until all active transactions have been rolled back," where the referenced "active transactions" include the claimed "actions," see Ganesh col. 1, ll. 19-32.

Klein teaches "*creating one or more other modified blocks to generate the transactional consistency in the read-only cache view*," see Fig. 10, step 168.

Klein does not teach "*creating a shadow cache configured to hold at least one modified block that has been physically redone and undone or that has been logically undone.*" Weems does, however, see Shadow Cache, "Small separate cache stores address of each evicted value," where, combined with Klein, the referenced shadow cache would be "configured to" hold modified blocks. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Weems teachings would have allowed Klein's method to keep recently evicted and reloaded references in volatile memory, see Weems Shadow Cache.

Klein does not teach "*determining a first log sequence number and a second log sequence number associated with the active transaction list, the first log sequence number and the second log sequence number specifying a set of transactions capable of being physically*

redone and undone or logically undone." Klein teaches rolling back transactions that are active or committed after the system change number of a requested read time, see Fig. 10. Ganesh teaches finer-grained temporal access by using a buffer refresh time (the time the data block in the buffer was last refreshed, see col. 7, ll. 26-33) and buffer commit time (the most recent commit time of the data block, see col. 7, ll. 34-48) to undo active and committed transactions, see Fig. 2. Natarajan teaches even finer-grained temporal access in the form of log sequence numbers, which are assigned to each log record, not just committed transactions, see 4.1. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to determine a first and second LSN associated with the active transactions because using Natarajan's LSNs as Ganesh's two buffer times would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1.

Klein does not teach "*comparing the log sequence number to both the first log sequence number and the second log sequence number*." Ganesh teaches comparing a snapshot time to a buffer refresh time and buffer commit time to rollback specific transactions, see Fig. 2. Natarajan teaches comparing LSNs, see 4.1. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to compare Natarajan's LSNs the way Ganesh compares the snapshot with buffer times because using Natarajan's LSNs as Ganesh's snapshot and buffer times would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1.

Klein does not teach "*and determining the one or more other modified blocks being stored in the shadow cache and not an original database if a shared cache overflows*." Weems teaches using a shadow cache, see Shadow Cache. Thus, it would have been obvious to one of

ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Weems teachings would have allowed Klein's method to keep recently evicted and reloaded references in volatile memory, see Weems Shadow Cache.

32. Klein teaches "*The method of claim 31, wherein the one or more other modified blocks are not written from the shadow cache to the original database*," see Fig. 10 and col. 9, ll. 5-17, "The retrieved data values from the rolled back transactions are provided (block 169)."

33. Klein teaches "*The method of claim 31, wherein the original database is used if the read-only cache view overflows a shared cache*," see Fig. 2 and col. 5, l. 65 – col. 6, l. 7, "Uncommitted transactions are stored separately from the data tables 36 in transaction table entries, as further described below with reference to FIG. 4, stored in rollback segments 35."

36. Klein does not teach "*The method of claim 31, wherein determining one or more other modified blocks comprises mapping the one or more other modified blocks in a table in the shadow cache, the table comprising a first column configured to maintain a block number of a read-only cache view block having an undo/redo record applied to it and a second column configured to maintain a block number allocated to temporarily store the one or more other modified blocks overflowing the shared cache and being stored in the shadow cache*." Klein does teach that "[i]n relational databases, data values are stored in tables organized according to a schema. The schema provides a mapping of the links between data sets and defines individual attributes. Related data values are stored in rows within the tables and each data value can be defined to store virtually any type of data object, including alphanumeric strings, numeric values, or binary data" (see col. 1, ll. 44-55). Weems teaches a shadow cache, see Shadow Cache. It would have been obvious to one of ordinary skill in the database art at the time of the invention

to use Klein's mapping table with Weems' shadow cache because it would have provided a way to access the data in the shadow cache, see Klein col. 1, ll. 44-55. The particular column headings claimed are obvious, as they are predictable variations of a mapping table, see KSR International Co. v. Teleflex Inc., 550 U.S. 398, 417, 82 USPQ2d 1385, 1396 (2007) ("If a person of ordinary skill can implement a predictable variation, § 103 likely bars its patentability").

37. Klein teaches "*The method of claim 31, further comprising marking the read-only cache view as closed upon termination of the read-only transaction*," see Fig. 10, "Return."

39. Klein teaches "*The method of claim 31, further comprising sharing the read-only cache view with other read-only transactions which start within a period of time following a start of the read- only transaction*," see col. 9, lines 5-17, "The retrieved data values from the rolled back transactions are provided (block 169)."

40. Klein teaches "*The method of claim 31, further comprising: detecting the read-only transaction*," see col. 8, lines 30-53, "FIG. 10 is a flow diagram showing a routine for performing a consistent read operation 160."

Klein teaches "*and adding a back link log record to a transaction log, the back link log record being configured to link to another log record in the transaction log that is associated with a write transaction, the adding being performed upon occurrence of a write operation*," see col. 2, lines 16-22, "Some database systems incorporate transaction logs which track and record all operations performed against the database. Log mining allows those operations which have effected the data to be reconstructed back into database statements."

42. Klein teaches "*A computer-readable storage medium having processor-executable instructions for performing the method of claim 31*," see col. 6, lines 24-34, "The various implementations of the source code and object and byte codes can be held on a computer-readable storage medium."

43. Klein teaches "*The method of claim 31, further comprising: downloading a set of processor-executable instructions for performing the method of claim 31; and storing the set of processor-executable instructions on a computer-readable storage medium configured to be accessed by at least one computer*," see col. 6, lines 24-34, "The various implementations of the source code and object and byte codes can be held on a computer-readable storage medium."

Claims 34-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over Klein et al., U.S. 6,631,374 ("Klein"), in view of Ganesh et al., U.S. 6,192,377 ("Ganesh"), in view of Weems, "Shadow Cache," University of Massachusetts, May 2004 ("Weems"), in view of Natarajan et al., "Log Sequence Numbers," University of Wisconsin, May 2003 ("Natarajan"), and in view of Hayashi et al., U.S. 5,715,447 ("Hayashi").

34. Klein does not teach "*The method of claim 31, further comprising using an allocation bitmap configured to indicate the modified block is to be temporarily stored in a temporary database*." Hayashi does, however, see Fig. 2, "bit map 30" and "overflow pages 31." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi's teachings would have allowed Klein's method to gain control of the overflow pages, see col. 4, ll. 8-18.

35. Klein does not teach "*The method of claim 34, further comprising deleting the modified block temporarily stored in the shadow cache by updating the allocation bitmap, the updating being performed upon completing the read-only transaction.*" Hayashi does, however, see col. 4, ll. 8-18, "any bit of the bit map 30 will be ON when a corresponding one of the overflow pages 31 is in use and OFF when the corresponding page is unused." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi's teachings would have allowed Klein's method to gain control of the overflow pages, see col. 4, ll. 8-18.

Claims 38 and 52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Klein et al., U.S. 6,631,374 ("Klein"), in view of Ganesh et al., U.S. 6,192,377 ("Ganesh"), in view of Weems, "Shadow Cache," University of Massachusetts, May 2004 ("Weems"), in view of Natarajan et al., "Log Sequence Numbers," University of Wisconsin, May 2003 ("Natarajan"), and in view of *The Authoritative Dictionary of IEEE Standards Terms, Seventh Edition*, IEEE Press, 2000 ("IEEE").

38. Klein does not teach "*The method of claim 37, further comprising: traversing the shared cache looking for a database block to purge when a new block allocation is needed in the shared cache; and purging the database block from the read-only cache view that has been marked as closed.*" IEEE does, however, see "garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database" and "cache (2) A small portion of high-speed memory used for temporary storage of frequently-

used data, instructions, or operands." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because IEEE's teachings would have allowed Klein's method and system to gain the ability to compact the contents of the database view, see "garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database."

52. Klein does not teach "*The system of claim 5 i, wherein the cache manager is configured to traverse the shared cache looking for the one or more blocks to purge, and is further configured to purge the one or more blocks from the read-only cache view if the one or more blocks from the read- only cache view have been marked as closed when a new block allocation is requested in the shared cache.*" IEEE does, however, see "garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database" and "cache (2) A small portion of high-speed memory used for temporary storage of frequently-used data, instructions, or operands." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because IEEE's teachings would have allowed Klein's method and system to gain the ability to compact the contents of the database view, see "garbage collection (B) A database reorganization technique in which the contents of a database are made more compact by physically deleting garbage such as records that have been deleted logically but remain physically in the database."

Claims 41 and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Klein et al., U.S. 6,631,374 ("Klein"), in view of Ganesh et al., U.S. 6,192,377 ("Ganesh"), in view of Weems, "Shadow Cache," University of Massachusetts, May 2004 ("Weems"), in view of Natarajan et al., "Log Sequence Numbers," University of Wisconsin, May 2003 ("Natarajan"), and in view of Raz, U.S. 5,701,480 ("Raz").

41. Klein does not teach "*The method of claim 40, further comprising using the back link log record to skip a portion of the transaction log that is irrelevant to undoing an uncommitted write transaction, wherein the back link log record is generated in the transaction log when there is an active read only transaction, if the read-only transaction must be undone.*" Raz does, however, see col. 62, ll. 3-17, "the computer 20 processes transactions using an 'undo' recovery mechanism that provides very fast recovery because only the effects of failed transactions must be undone." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Raz's teachings would have allowed Klein's method and system to gain greater efficiency by not undoing redundant or unnecessary transactions, see Raz col. 62, ll. 3-17.

55. Klein does not teach "*The system of claim 44, wherein the log manager is configured to use a back link log record to skip a portion of the transaction log that is not associated with undoing the write operation, wherein the back link log record is generated in the transaction log when there is an active read only transaction.*" Raz does, however, see col. 62, ll. 3-17, "the computer 20 processes transactions using an 'undo' recovery mechanism that provides very fast recovery because only the effects of failed transactions must be undone." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine

the teachings of the cited references because Raz's teachings would have allowed Klein's method and system to gain greater efficiency by not undoing redundant or unnecessary transactions, see Raz col. 62, ll. 3-17.

Claims 44-51 and 53-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Klein et al., U.S. 6,631,374 ("Klein"), in view of Ganesh et al., U.S. 6,192,377 ("Ganesh"), in view of Weems, "Shadow Cache," University of Massachusetts, May 2004 ("Weems"), in view of Natarajan et al., "Log Sequence Numbers," University of Wisconsin, May 2003 ("Natarajan"), in view of Hayashi et al., U.S. 5,715,447 ("Hayashi"), and in view of DeWitt, Jr. et al., U.S. 7,093,081 ("DeWitt").

44. Klein teaches "*A database system configured to restore a database to a consistent version supporting read-only uses, comprising*," see col. 2, ll. 38-50, "[Klein] provides a system and method for temporally accessing data values in a database as of a requested query time."

Klein teaches "*a computer having a processor and memory*," see Fig. 1, Server 10.

Klein teaches "*a log manager module configured to manage a transaction log of the database system*," see col. 2, ll. 15-22, "Some database systems incorporate transaction logs which track and record all operations performed against the database."

Klein does not teach "*a cache manager module configured to manage a shared cache that stores one or more database blocks in a memory of the database system and configured to create a write view of the database in the shared cache supporting read and write uses of the database*." Hayashi does, however, see Fig. 4 and col. 1, ll. 30-41, "When a transaction accesses the database for some data through the database management system, the data may already be in

a buffer shared by transactions due to another transaction that previously accessed the same data, or the data must be transferred from the database to the shared buffer." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi's teachings would have allowed Klein's method to gain a common means for storing frequently used database blocks, see col. 1, ll. 30-41. Klein teaches "*and a read-only cache view of the database using the transaction log of the database, the read-only cache view being created in response to a read-only transaction of the database, the read-only cache view comprising the one or more database blocks of the shared cache that record a view of a version of the database at a time*," see Fig. 10, col. 2, ll. 16-22, "Some database systems incorporate transaction logs which track and record all operations performed against the database. Log mining allows those operations which have effected the data to be reconstructed back into database statements," col. 8, ll. 30-53, "FIG. 10 is a flow diagram showing a routine for performing a consistent read operation 160… First, the relative database block is retrieved (block 161) from the persistent storage 22. Next, each interested transaction entry (ITE) 51 (shown in FIG. 3) is iteratively processed (blocks 162-169) to logically reconstruct the database 23 as of the requested query time… For each ITE 51, transactions are rolled back in an ordered manner." Klein does not teach "*wherein the cache manager is configured to use a non-transactional database to store the one or more database blocks that overflow the shared cache during use of the read-only cache view by the read-only transaction.*" DeWitt does, however, see col. 4, ll. 3-21, "However, if there is not sufficient space in the cache to store the block of instructions/data that is to be reloaded, then the block of instructions/data, or at least the overflow portion of the block of instructions/data, is loaded into a reserved portion of

cache," where the claimed "temporary database" is the referenced "reserved portion of cache." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because DeWitt's teachings would have allowed Klein's method to store blocks of data that overflow the cache, see col. 4, ll. 3-21. Klein does not teach "*wherein the steps to create the read-cache view include determining a first log sequence number and second log sequence number from the transaction log*." Klein teaches rolling back transactions that are active or committed after the system change number of a requested read time, see Fig. 10. Ganesh teaches finer-grained temporal access by using a buffer refresh time (the time the data block in the buffer was last refreshed, see col. 7, ll. 26-33) and buffer commit time (the most recent commit time of the data block, see col. 7, ll. 34-48) to undo active and committed transactions, see Fig. 2. Natarajan teaches even finer-grained temporal access in the form of log sequence numbers, which are assigned to each log record, not just committed transactions, see 4.1. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to determine a first and second LSN associated with the active transactions because using Natarajan's LSNs as Ganesh's two buffer times would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1. Klein teaches "*loading a modified block including a log sequence number into the read-only cache view*," see Fig. 10, step 161. Klein does not teach "*comparing the log sequence number to the first log sequence number and the second log sequence number*." Ganesh teaches comparing a snapshot time to a buffer refresh time and buffer commit time to rollback specific transactions, see Fig. 2. Natarajan teaches comparing LSNs, see 4.1. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention

to compare Natarajan's LSNs the way Ganesh compares the snapshot with buffer times because using Natarajan's LSNs as Ganesh's snapshot and buffer times would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1. Klein teaches "*and performing physical redo operations or physical undo operations based upon the comparing as the modified block is loaded into the read-only cache view*," see Fig. 10, steps 166-67, where the claimed "comparing," when combined with Ganesh and Natarajan, would be the referenced step 166. Klein does not teach "*wherein the physical redo operations or the physical undo operations are not performed on the modified block when the log sequence number is less than the fist log sequence number*." Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches using the current version of the data block if the snapshot (i.e. the "log sequence number" when combined with Natarajan) is less than the buffer refresh time (i.e. the "first log sequence number" when combined with Natarajan), see Fig. 2, steps 206 and 216. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to use Natarajan's LSNs as Ganesh's snapshot and buffer times and to use Ganesh's time comparisons because doing so would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1. Klein does not teach "*and the physical redo operations are performed on the modified block when the log sequence number is greater than the first log sequence number and less than the second log sequence number*." Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches using another version of the data block if the snapshot (i.e. "log sequence number") is greater than the buffer refresh time (i.e. "first log sequence number"), see Fig. 2, steps 206 and 208. Thus, it would have been obvious to one of

ordinary skill in the database art at the time of the invention that Ganesh's step 208 could be redoing the transaction instead of using a different version, since it is a predictable variation, see KSR International Co. v. Teleflex Inc., 550 U.S. 398, 417, 82 USPQ2d 1385, 1396 (2007) ("If a person of ordinary skill can implement a predictable variation, § 103 likely bars its patentability"). Klein does not teach "*and the physical undo operations are performed on the modified block when the log sequence number is greater than the second log sequence number.*" Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches undoing changes when the snapshot (i.e. "log sequence number") is greater than the data buffer commit time (i.e. "second log sequence number") and the transaction is active, see Fig. 2, steps 210, 214, and 212. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to use Natarajan's LSNs as Ganesh's snapshot and buffer times and to use Ganesh's time comparisons because doing so would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1.

Klein teaches "*and a transaction manager module configured to logically undo one or more transactions on demand, the one or more transactions having begun but having yet to commit upon starting the read-only transaction in order to construct the read-only cache view comprising a transactionally consistent prior version of the database,*" see Fig. 10 and col. 8, ll. 30-53, "For each ITE 51, transactions are rolled back in an ordered manner… Thus, if the transaction referenced by the transaction identifier xid is active (block 163), the entire transaction is rolled back to logically undo the transaction (block 164)." Klein further teaches "*performing the read-only transaction using the read-only cache view without blocking performance of the one or more transactions involving a write operation using a write view of the database, and*

returning a result associated with the read-only transaction," see Fig. 10, col. 9, ll. 5-17, "The retrieved data values from the rolled back transactions are provided (block 169)," and col. 7, l. 66 – col. 8, l. 8, "As would be readily recognized by one skilled in the art, a typical database engine could be concurrently processing multiple transaction operations at any given time. The described invention implicates read operations and does not directly effect the processing of other forms of database transactions."

45. Klein teaches "*The system of claim 44, wherein a database block associated with the read-only cache view is not written from the shared cache to the given database during occurrence of the read-only transaction*," see Fig. 10 and col. 9, ll. 5-17, "The retrieved data values from the rolled back transactions are provided (block 169)."

46. Klein does not teach "*The system of claim 44, wherein the cache manager stores a database block that overflows the shared cache in a temporary database*." DeWitt does, however, see col. 4, ll. 3-21, "However, if there is not sufficient space in the cache to store the block of instructions/data that is to be reloaded, then the block of instructions/data, or at least the overflow portion of the block of instructions/data, is loaded into a reserved portion of cache," where the claimed "temporary database" is the referenced "reserved portion of cache." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because DeWitt's teachings would have allowed Klein's method to store blocks of data that overflow the cache, see col. 4, ll. 3-21.

47. Klein does not teach "*The system of claim 46, wherein the non-transactional database is used if the read-only cache view overflows the shared cache*." DeWitt does, however, see col. 4, ll. 3-21, "However, if there is not sufficient space in the cache to store the

block of instructions/data that is to be reloaded, then the block of instructions/data, or at least the overflow portion of the block of instructions/data, is loaded into a reserved portion of cache," where the claimed "temporary database" is the referenced "reserved portion of cache." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because DeWitt's teachings would have allowed Klein's method to store blocks of data that overflow the cache, see col. 4, ll. 3-21.

48. Klein does not teach "*The system of claim 44, wherein the cache manager maintains an allocation bitmap identifying the one or more database blocks being temporarily stored in a temporary database.*" Hayashi does, however, see Fig. 2, "bit map 30" and "overflow pages 31." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi's teachings would have allowed Klein's method to gain control of the overflow pages, see col. 4, ll. 8-18.

49. Klein does not teach "*The system of claim 48, wherein the cache manager is configured to delete the one or more blocks from the temporary database by updating the allocation bitmap.*" Hayashi does, however, see col. 4, ll. 8-18, "any bit of the bit map 30 will be ON when a corresponding one of the overflow pages 31 is in use and OFF when the corresponding page is unused." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi's teachings would have allowed Klein's method to gain control of the overflow pages, see col. 4, ll. 8-18.

50. Klein does not teach "*The system of claim 44, wherein the cache manager stores a mapping to the one or more blocks overflowing the shared cache in a table of the non-*

transactional database including a first column configured to maintain a block number of a read-only cache view block having an undo/redo record and a second column configured to maintain a block number allocated to temporarily store the one or more blocks overflowing the shared cache." Hayashi does, however, see col. 5, lines 30-46, "The database contains page data and a table showing relationships between page numbers and locations on the disk." Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to combine the teachings of the cited references because Hayashi's teachings would have allowed Klein's method to gain control of the overflow pages, see col. 4, lines 8-18.

51. Klein teaches "*The system of claim 44, wherein the cache manager is configured to mark the read-only cache view as closed upon termination of the read-only transaction*," see col. 9, lines 5-17, "The retrieved data values from the rolled back transactions are provided (block 169)."

53. Klein teaches "*The system of claim 44, wherein the cache manager is configured to share the read-only cache view created for the read-only transaction with other read-only transactions that are configured to start within a period of time following a start of the read-only transaction*," see col. 9, lines 5-17, "The retrieved data values from the rolled back transactions are provided (block 169)."

54. Klein teaches "*The system of claim 44, wherein the log manager is configured to detect the read-only transaction, and further configured to add a back link log record to the transaction log that is configured to link together two or more log records of the transaction log that are associated with a write transaction to be logically undone*," see col. 8, lines 30-53, "FIG. 10 is a flow diagram showing a routine for performing a consistent read operation 160"

and col. 2, lines 16-22, "Some database systems incorporate transaction logs which track and record all operations performed against the database. Log mining allows those operations which have effected the data to be reconstructed back into database statements."

Response to Arguments

As per applicant's argument that the prior art of record does not teach "wherein the physical redo operations or the physical undo operations are not performed on the modified block when the log sequence number is less than the fist log sequence number, and the physical redo operations are performed on the modified block when the log sequence number is greater than the first log sequence number and less than the second log sequence number, and the physical undo operations are performed on the modified block when the log sequence number is greater than the second log sequence number" as recited in claims 31 and 44, the examiner respectfully disagrees.

Klein teaches rolling back transactions that are active or committed after the system change number of a requested read time, see Fig. 10. Ganesh teaches finer-grained temporal access by using a buffer refresh time (the time the data block in the buffer was last refreshed, see col. 7, ll. 26-33) and buffer commit time (the most recent commit time of the data block, see col. 7, ll. 34-48) to undo active and committed transactions, see Fig. 2. Natarajan teaches even finer-grained temporal access in the form of log sequence numbers, which are assigned to each log record, not just committed transactions, see 4.1. Natarajan also teaches comparing LSNs to determine which transactions to redo, see 4.1. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to use Natarajan's LSNs as Ganesh's

snapshot and buffer times because doing so would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1.

Specifically:

Klein does not teach "*wherein the physical redo operations or the physical undo operations are not performed on the modified block when the log sequence number is less than the fist log sequence number*." Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches using the current version of the data block if the snapshot (i.e. the "log sequence number" when combined with Natarajan) is less than the buffer refresh time (i.e. the "first log sequence number" when combined with Natarajan), see Fig. 2, steps 206 and 216. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to use Natarajan's LSNs as Ganesh's snapshot and buffer times and to use Ganesh's time comparisons because doing so would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1.

Klein does not teach "*and the physical redo operations are performed on the modified block when the log sequence number is greater than the first log sequence number and less than the second log sequence number*." Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches using another version of the data block if the snapshot (i.e. "log sequence number") is greater than the buffer refresh time (i.e. "first log sequence number"), see Fig. 2, steps 206 and 208. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention that Ganesh's step 208 could be redoing the transaction instead of using a different version, since it is a predictable variation, see KSR International Co. v. Teleflex Inc., 550 U.S. 398, 417, 82 USPQ2d 1385, 1396 (2007) ("If a

person of ordinary skill can implement a predictable variation, § 103 likely bars its patentability").

Klein does not teach "*and the physical undo operations are performed on the modified block when the log sequence number is greater than the second log sequence number.*" Natarajan teaches comparing LSNs to determine whether to redo an update, see 4.1. Ganesh teaches undoing changes when the snapshot (i.e. "log sequence number") is greater than the data buffer commit time (i.e. "second log sequence number") and the transaction is active, see Fig. 2, steps 210, 214, and 212. Thus, it would have been obvious to one of ordinary skill in the database art at the time of the invention to use Natarajan's LSNs as Ganesh's snapshot and buffer times and to use Ganesh's time comparisons because doing so would have provided Klein's method with more fine-grained temporal access, see Ganesh col. 4, ll. 42-64 and Natarajan 4.1.

Thus, the 35 U.S.C. 103 rejection of claims 31-55 is maintained.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aaron Sanders whose telephone number is 571-270-1016. The examiner can normally be reached on M-F 9:00a-4:00p.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on 571-272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168

/Aaron Sanders/
Examiner, Art Unit 2168
27 October 2009